

2. Input Devices

2.1 Pointing Device Input

2.1.1 The Pointer

The pointing device is used to move a pointer on the screen, select objects on which the pointer is placed, and manipulate the objects directly. The pointing device is associated with a single pointer which provides a graphical representation of the location of the pointing device on the screen. The pointer may change shape depending on where it is located. The hotspot of the pointer (i.e., the active point) indicates the precise location where pointing device operations occur (e.g., the object that will be selected when users execute a select action). The location of the hotspot does not move as the pointer changes from one shape to another.

The hotzone is the area that the hotspot must be within to be considered on an object. The hotzone is defined by the boundary of an object; in some cases (e.g., radio buttons and check boxes), the hotzone includes both the graphic for an object and its label. The application does not diverge from the pointing device model defined here (i.e., the hotspot of the pointer indicates the locus of user input) or modify the size of the hotzone for any interface component.

2.1.2 Pointer Shapes

The shape of the pointer provides feedback about the function of the area on which the pointer is located. Figure 2-1 lists some of the common pointer shapes defined by Motif and MS Windows. The application uses these shapes whenever the pointer is over an area where the function applies. If the application redefines the pointer shape, it does so only when the pointer is in an application window. The upper-left-pointing arrow is the general-purpose pointer for object selection in most windows. The pointer is assumed to have this shape, unless otherwise indicated.





















<u>Motif</u>	<u>MS Windows</u>
<u>Selection Pointers</u>  Arrow pointer (used to select objects)  I-beam pointer (used to select text)  Sighting pointer (used to make fine position selections)	<u>Selection Pointers</u>  Arrow pointer (used to select objects)  I-beam pointer (used to select text)
<u>Move and Resize Pointers</u>  Move pointer (indicates a move operation in progress)  Sizing pointers (indicate positions and directions for resizing an area)	<u>Move and Resize Pointers</u>  Resize column & row pointers (indicate direction for resizing a column or row)  Sizing pointers (indicate directions for resizing a sizable edge)
<u>Working/Caution Pointers</u>  Watch pointer (indicates that an operation is in progress)  Caution pointer (indicates that an action is expected in another area before input can be made in the current area)  Cannot pointer (indicates that a drag icon is over an invalid drop zone)	<u>Working/Caution Pointers</u>  Hourglass pointer (indicates that an operation is in progress)  "No" pointer (indicates that a drop action is not allowed at this location)
<u>Help Pointers</u>  Help pointer (indicates context-sensitive help)	<u>Help Pointers</u>  Help pointer (indicates context-sensitive help)
<u>Other Pointers</u>  Out-of-range pointer (indicates that the pointer has moved outside an application area)  Menu pointer (indicates a pending menu action)	<u>Other Pointers</u>  Zoom pointer (magnifies an area of a window)  Split pointers (splits a window horizontally or vertically)

Figure 2-1. Common pointer shapes in Motif and MS Windows.

The application does not create new pointer shapes for functions that already have a shape and does not use existing shapes for functions they were not designed to represent. If the application creates a new pointer shape for functionality not listed in figure 2-1, it is easy to see (e.g., has high contrast with the background, does not obscure other information on the screen), with a hotspot that is obvious and easy to locate. In addition, the new shape provides a hint to its purpose and is not easily confused with other objects on the screen.

2.1.3 Pointing Device Buttons

The buttons on the pointing device are used to perform functions related to object manipulation. These functions include selecting objects, transferring objects, and displaying pop-up menus. The manner in which pointing device buttons are assigned to functions depends on the number of buttons available and whether selection and transfer functions are integrated or separate. This document assumes that a pointing device can have two or three buttons and that selection and transfer are integrated (i.e., a single button is used to perform both functions).¹

The application uses the left button (i.e., BLeft) on the pointing device to select and transfer objects, and the right button (i.e., BRight) to display pop-up menus;² the middle button (i.e., BMiddle), if available, is used to provide redundant or short-cut access to functions available elsewhere in the interface.³

Motif Only: If the application supports separate select and transfer functions, BLeft is used to select objects and BMiddle is used to transfer objects.

The following actions can be performed with the pointing device:

Press	Press and hold down a button.
Release	Release a button after it has been pressed.
Click	Press and release a button without moving the pointing device.
Double click	Press and release a button twice in rapid succession without moving the pointing device.
Move	Move the pointing device without pressing any buttons.
Drag	Move the pointing device while pressing a button.

2.2 Keyboard Input

2.2.1 Text Entry

A keyboard includes alphanumeric keys, punctuation and symbol keys, and special keys such as SPACE, RETURN, BACKSPACE, DELETE, and TAB. SPACE inserts a space during text entry, while RETURN inserts a carriage return in multi-line text. If no text has been selected, BACKSPACE deletes the character to the left of the text cursor (i.e., backward deletion) and DELETE deletes the character to the right of the text cursor (i.e., forward deletion). If text has

¹ MS Windows supports integrated select and transfer functions, while the CDE version of Motif gives preference to this model. Previous versions of Motif call for selection and transfer to be separate.

² Unless otherwise indicated, the remainder of this document assumes this mapping of functions to pointing device buttons.

³ Motif refers to the left, middle, and right buttons on a three-button pointing device as buttons 1, 2, and 3, respectively. MS Windows refers to the left and right buttons on a two-button pointing device as buttons 1 and 2, respectively; if the pointing device has three buttons, the right button is button 2 and the middle button is button 3. The specifications presented here refer to BLeft, BMiddle, and BRight in order to minimize confusion when identifying individual pointing device buttons.

been selected, both BACKSPACE and DELETE delete the selection. TAB inserts a tab or moves to the next tab stop in multi-line text.

Two modes are available for text entry, with INSERT used to toggle between the modes.⁴ In insert mode, when users begin to type, the new character is added at the position of the text cursor. In replace mode, when users begin to type, the new character replaces the one under the text cursor. The text cursor is a vertical bar (|) in insert mode and a shaded rectangle in replace mode.⁵

2.2.2 Fixed Function Keys

Motif and MS Windows use fixed function keys (i.e., each key has only one predefined function associated with it) to perform navigation and selection from the keyboard. Appendix A identifies the keys to be used for these functions, while appendix B maps the keys to the keyboards for several DII hardware platforms. The application uses the keys listed in appendix A to provide redundant access to navigation and selection from the keyboard. Subsequent sections of this document identify the keyboard functions to be supported for each interface component, while the implementation of variable function keys is addressed later in this section.

2.2.3 Mnemonics

A mnemonic is an alphanumeric character in the label of a menu or control that, when used in combination with ALT, navigates to the component and selects or activates it. If the component already has focus, typing the mnemonic selects or activates it. If the component is unavailable (i.e., its label is dimmed), the mnemonic has no effect (i.e., executes no action). The mnemonic is included in the label of the component and underlined, as shown in figure 2-2. The mnemonic is not case-sensitive; users can type it in either upper or lower case.

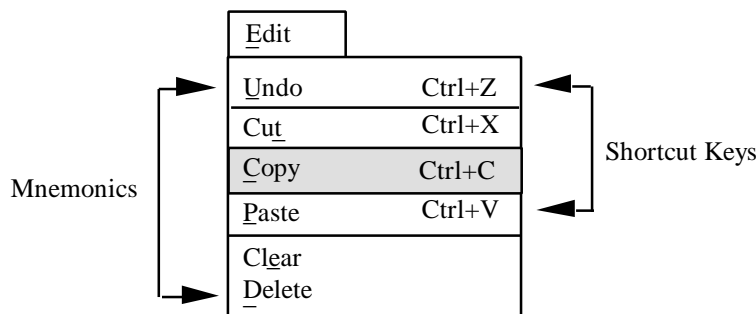


Figure 2-2. Example mnemonics and shortcut keys in MS Windows.

Appendix C lists the mnemonics for common actions used in Motif and MS Windows. If the application implements mnemonics for any of these actions, it uses the key listed in this appendix; exceptions can occur in order to provide a unique character within a set of components. If application-specific mnemonics are created, the mnemonic is the first character of the label; a

⁴ MIL-STD 1472E indicates that overstrike mode should not be used for data entry.

⁵ Previous versions of Motif use the same text cursor shape in both insert and replace mode.

distinctive consonant in the label is preferred over vowels. If the mnemonic is not the first character, it is the last character of the label or the first character of the second word (if there is more than one word) or a sequential number that is assigned to the label. If the mnemonic does not appear in the label, it is placed in parentheses following the text.

2.2.4 Shortcut Keys

A shortcut key consists of one or more keys that, when pressed, execute the action of a menu or control (e.g. a push button) in the window with input focus. If the component is unavailable or the window containing the component is minimized, the shortcut key has no effect. The name of the shortcut key includes a plus sign (e.g., Ctrl+X in figure 2-2) to indicate the keys to be pressed at the same time. The first letter of each key in the name is capitalized.

Appendix C lists the shortcut keys for common actions used in Motif and MS Windows. If the application implements shortcut keys for any of these actions, it uses the keys listed in this appendix. If application-specific shortcut keys are created, they have the form “modifier+character,” where the modifier is ALT, CTRL, SHIFT or a combination of these keys, and the character is an alphanumeric or special key on the keyboard. The application assigns SHIFT+key(s) combinations to actions that extend or are complementary to the actions of the key(s) used without SHIFT. The application uses CTRL+key(s) combinations for infrequent actions or for actions that represent larger-scale versions of the actions assigned to the unmodified key(s).

2.2.5 Variable Function Keys

If the application implements variable function keys, they are displayed as “soft” keys in a window, as shown in Figure 2-3. Soft keys can take on different meanings depending on the current state of the application; when the meaning of a key changes, its label in the window is modified to reflect the action that will be executed if the key is used. If the application changes the actions assigned to a set of soft keys, it limits the actions to no more than two per key, with additional actions available only if required to perform the operational task. In addition, the application defines keys that return to the previous level and to the set of base-level actions.

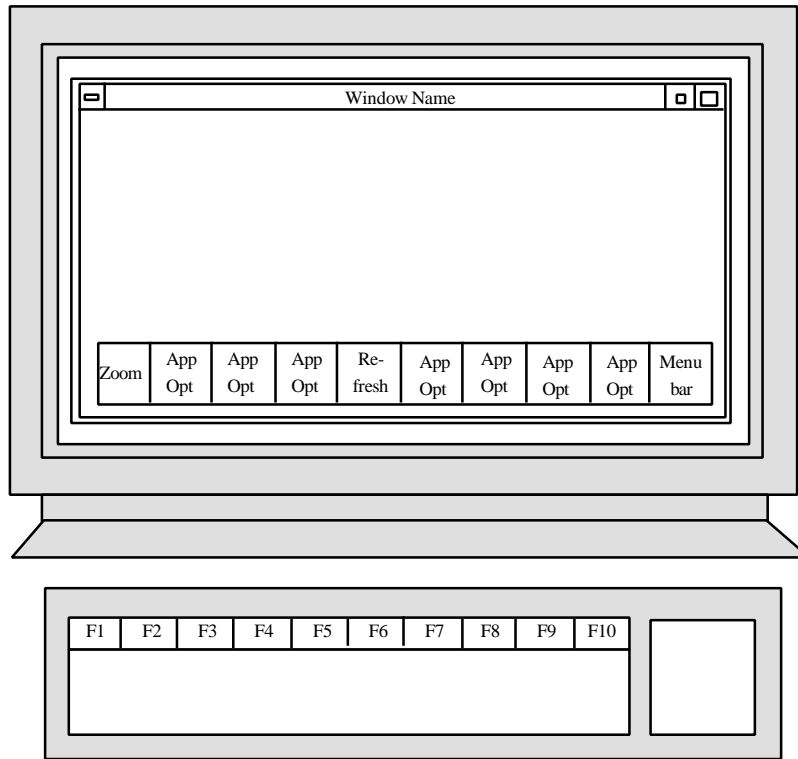


Figure 2-3. Example soft keys in a window.

If the application implements soft keys, users can execute the actions only when the window displaying the keys has input focus. In addition, the actions mapped to soft keys do not conflict with the key mappings in appendix A. The keys are grayed out (to indicate their unavailability) when the window does not have focus. Developers should expect that users will be working in multiple applications simultaneously, and the advantages associated with using variable function keys may no longer apply when the application is one of many available to users.

2.3 Alternate Input Devices

Developers considering the use of a hardware configuration that includes an input device other than a mouse or trackball should submit their requests to the appropriate configuration management board for approval prior to implementation. If the use of an alternate input device is approved, the manner in which users interact with the device (e.g., for navigation and selection) is consistent with the interaction models presented in section 3 of this document.⁶

⁶ Guidelines on user interface design using pen input are provided in [The Windows Interface Guidelines for Software Design](#).